

Compilador Gráfico

Nombre: Javi Granados
Curso: DAM - DAW 1C

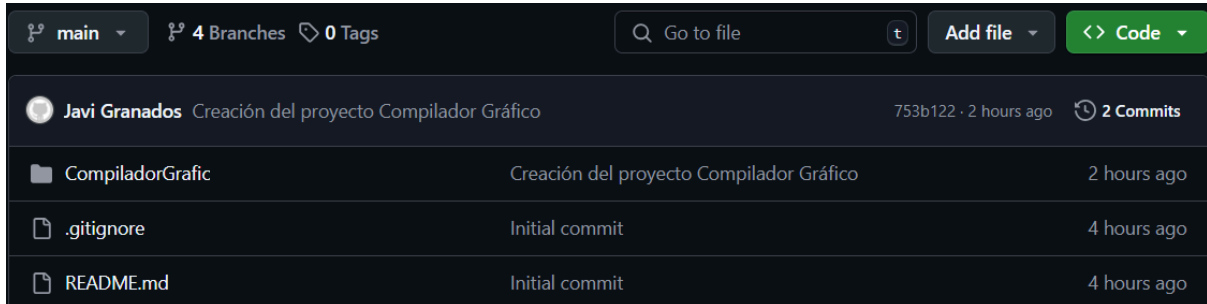
Índice

Trabajo con ramas git.....	3
Creación del proyecto.....	3
Ramas del repositorio.....	3
Trabajo en distintas ramas.....	4
Combinar cambios de diferentes ramas.....	5
Pull Request.....	5
Git Rebase.....	8
Combinación de Entorno Gráfico a Develop.....	9
Subida final del proyecto.....	10

Trabajo con ramas git

Creación del proyecto

En primer lugar, después de la creación del repositorio remoto, se crea el proyecto en el cual se va a trabajar y sin haber hecho cambios, se sube a la rama main del repositorio para guardar su estructura.



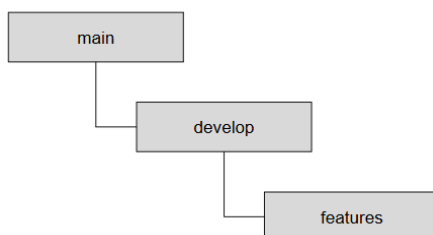
Ramas del repositorio

Una vez hecho el primer paso, crearemos una rama develop, esta sirve para subir las diferentes partes del proyecto una vez finalizadas.

Ahora si, se divide el proyecto en distintas partes y se crea una rama por cada una, por ejemplo si los participantes del equipo se dividen el trabajo, se puede crear una rama para cada integrante del equipo. Yo por ejemplo, he dividido mi proyecto en dos ramas, feature compilador, que contiene la clase compilador y una carpeta con las clases pila y tabla lista y feature entorno gráfico que contiene todo el entorno gráfico.

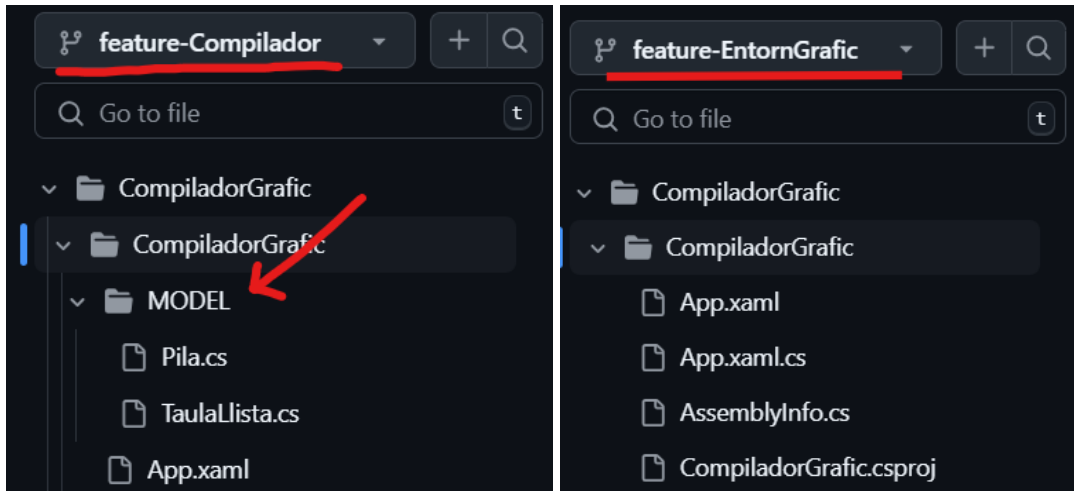
Esta dinámica de trabajo, permite a diferentes personas trabajar en su parte del proyecto, cuando esta está acabada, se sube a develop con un pull request. Si todas las partes están acabadas, se puede probar el correcto funcionamiento en develop, si este funciona, se hace un último pull request a main y se acaba la fase de desarrollo.

Estructura de ramas:



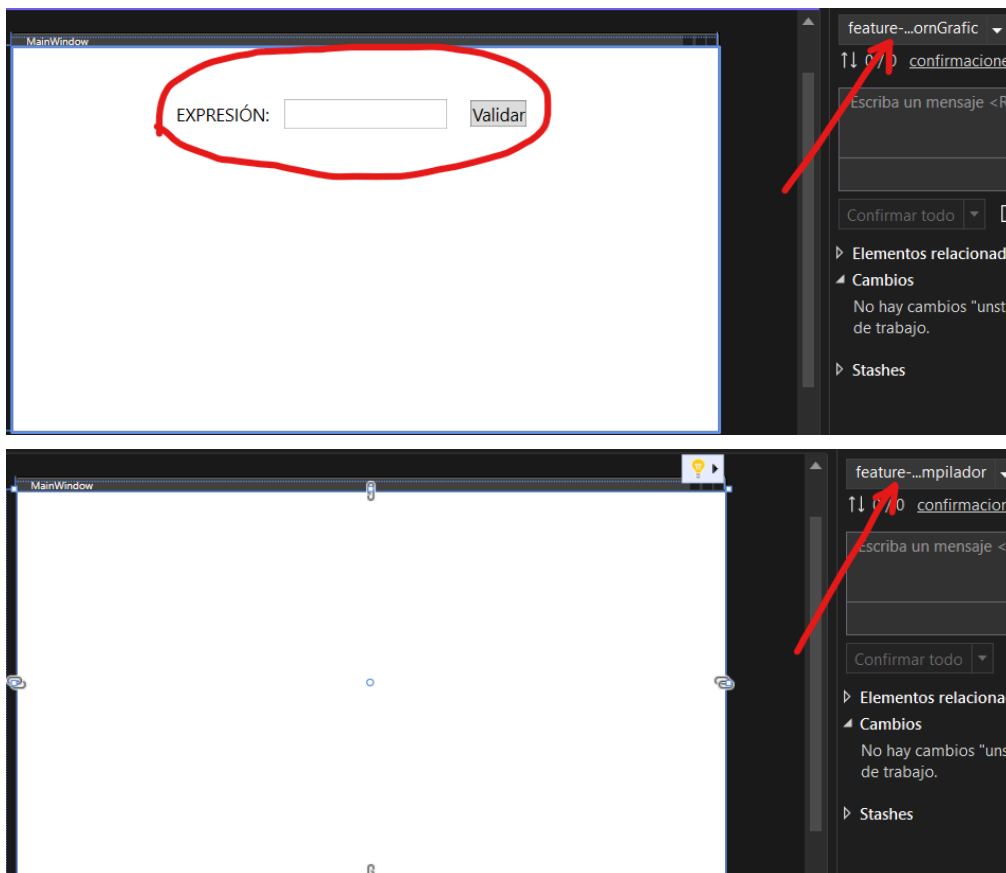
Trabajo en distintas ramas

Para no crear conflictos, el trabajo de cada rama es único y no se puede ver desde otras ramas. Por ejemplo, yo he hecho cambios en feature Compilador y desde feature Entorno Gráfico no veo estos cambios:



En este ejemplo, se puede ver como la carpeta MODEL de la rama Compilador no se ve en la rama entorno gráfico.

Pasa lo mismo en el programa, que depende de la rama en la que esté, se ven unos cambios o no:



Combinar cambios de diferentes ramas

Pull Request

Una vez acabados los cambios en una rama, ya podemos subirlos a develop, pero para esto, debemos hacer un pull request. Esto sirve para revisar los cambios antes de combinarlos en una rama superior y para que las demás personas que trabajan en el proyecto puedan ver estos cambios, si todo está correcto y no hay que hacer ninguna modificación, se acepta el pull request y los cambios se combinan con la rama de destino.

Con que ya tengo acabada la clase compilador de mi programa, ya puedo subir los cambios a la rama develop. Una vez hecho el último commit y push de feature compilador, desde la rama develop creo la pull request.

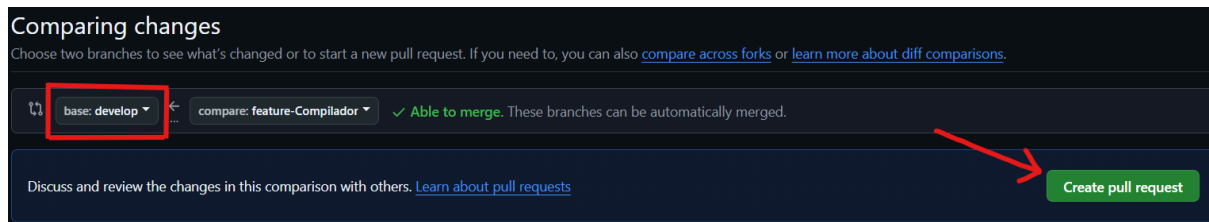
The screenshot shows the GitHub 'Commits' page for the 'feature-Compila...' branch. It lists two commits by Javi Granados: 'Método EstructuraNotPolaca cambiada lista salida de <char> a <string> y hechos método Resolver. --> Compilador Acabado' (committed 1 hour ago) and 'Método EstructuraNotPolaca para la determinar la estructura de una expresión en Notación Polaca inversa, usa el método Desapilar' (committed 14 hours ago). Below the commits, the navigation bar includes 'Pull requests', 'Actions', 'Projects', 'Wiki', 'Security', 'Insights', and 'Settings'. A message states: 'Label issues and pull requests for new contributors. Now, GitHub will help potential first-time contributors discover issues labeled with good first issue.' A red arrow points from this message to the 'New pull request' button in the bottom right corner. The bottom bar also shows 'Labels: 9' and 'Milestones: 0'.

Selecciono la rama que me interesa:

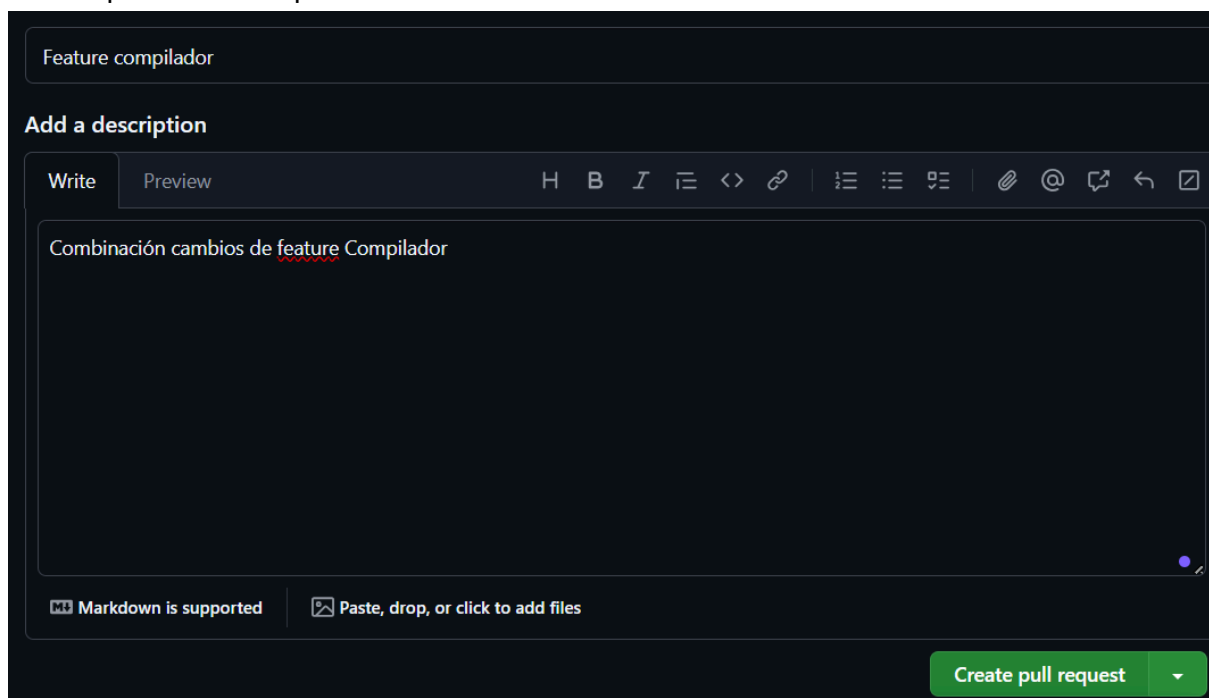
The screenshot shows the 'Compare and review just about anything' page. It states: 'Branches, tags, commit ranges, and time ranges. In the same repository and across forks.' Below this, there is a table titled 'Example comparisons'.

Example comparisons	
feature-Compilador	1 hour ago
feature-EntornGrafic	yesterday
develop	yesterday
main@{1day}...main	24 hours ago

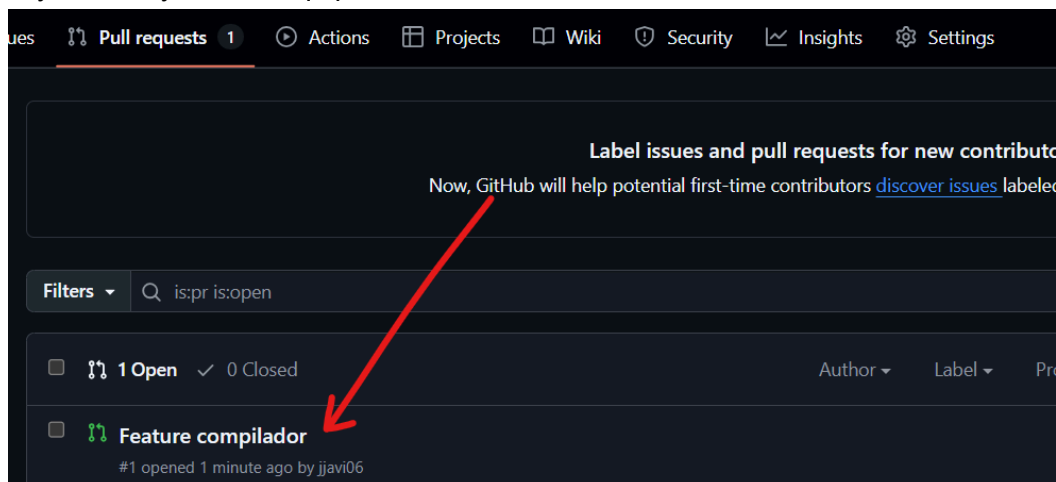
Con la rama seleccionada, sale una ventana para comparar los cambios, aquí es importante seleccionar la rama de destino correcta, una vez hecho esto, creamos el pull request.



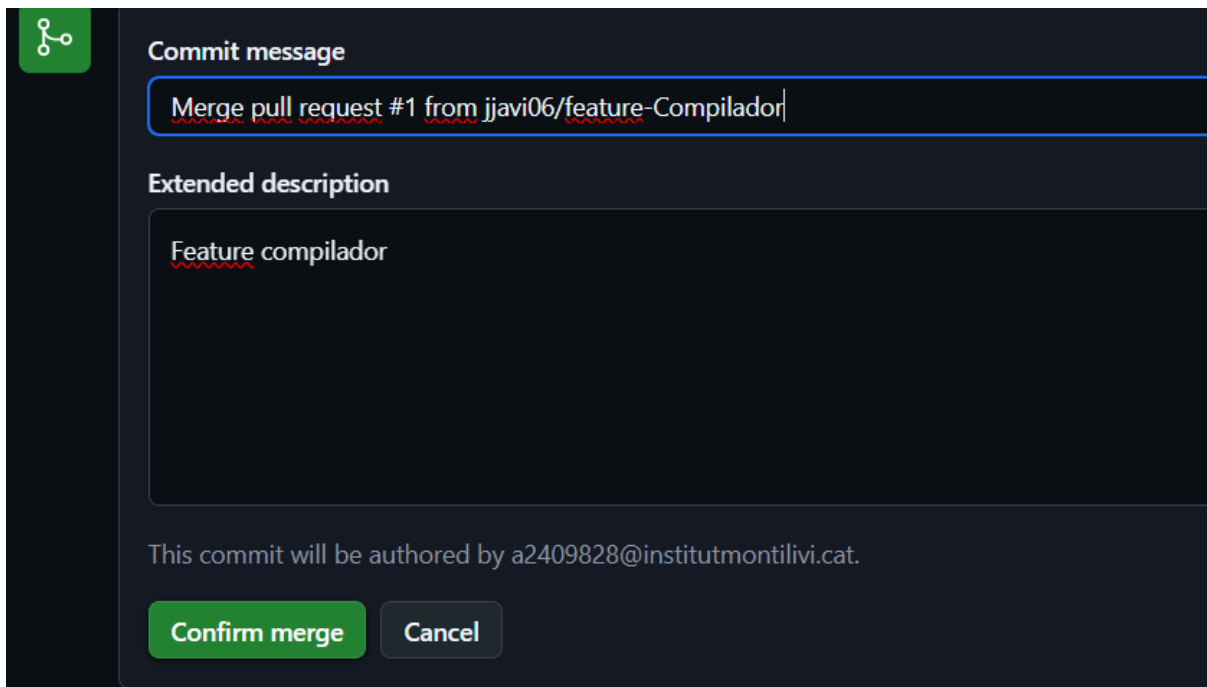
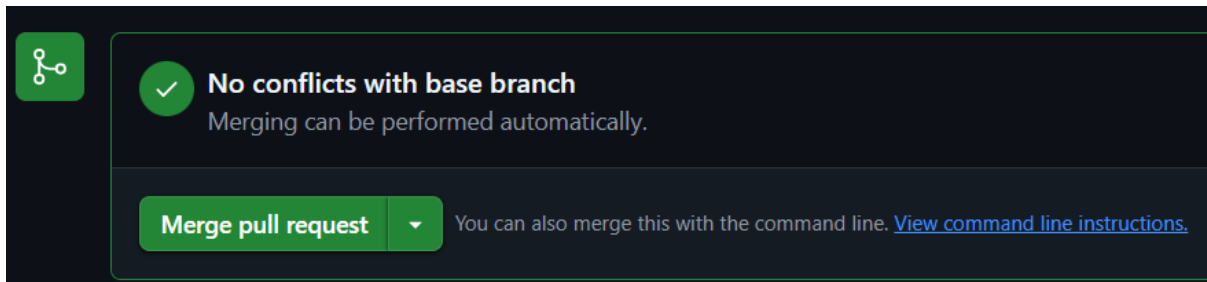
Haber hecho este paso, no significa que los cambios ya se hayan combinado, ahora falta aceptar estos cambios, normalmente los suele aceptar alguien del equipo con más responsabilidades ya que hay que asegurarse de que no hayan errores. Pero para publicar estos cambios, aún hay un paso previo y es añadir el comentario que queramos y si hace falta se pueden editar parámetros de la solicitud.



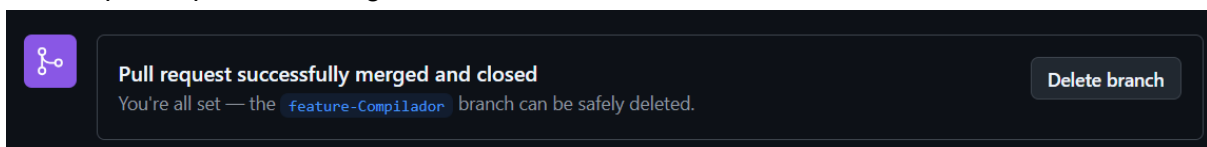
Y ahora sí que la solicitud de combinación de cambios está disponible para todos y si no hay errores y todo el equipo está de acuerdo, se combinan los cambios.



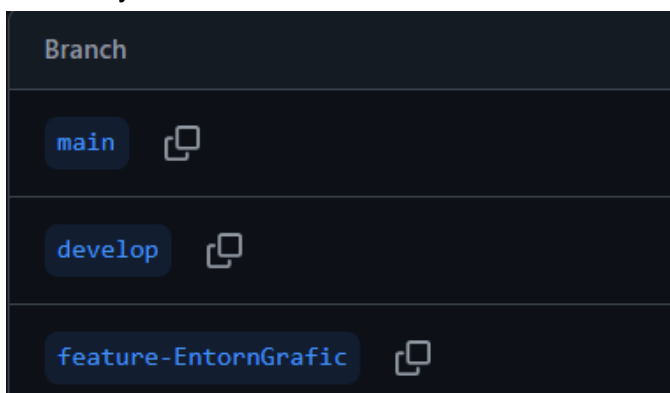
Como se puede ver no hay conflictos, entonces combinamos los cambios:



Y ya que los cambios ya están combinados, podemos borrar la rama feature Compilador, con la opción que el mismo github nos ofrece:



La rama ya no sale:



Git Rebase

Ahora lo que me interesa, es combinar los cambios que he subido anteriormente a develop, en la feature Entorno Gráfico para poder trabajar con el compilador y las demás clases dentro del entorno gráfico.

Para hacer esto, desde git bash haré un git rebase.

Primero, me coloco en la rama feature Entorno gráfico:

```
Usuario@DESKTOP-SGELNL5 MINGW64 ~/source/repos/WS8-BRANQUES-GIT (develop)
$ git checkout feature-EntornGrafic
Switched to branch 'feature-EntornGrafic'
Your branch is up to date with 'origin/feature-EntornGrafic'.

Usuario@DESKTOP-SGELNL5 MINGW64 ~/source/repos/WS8-BRANQUES-GIT (feature-EntornGrafic)
```

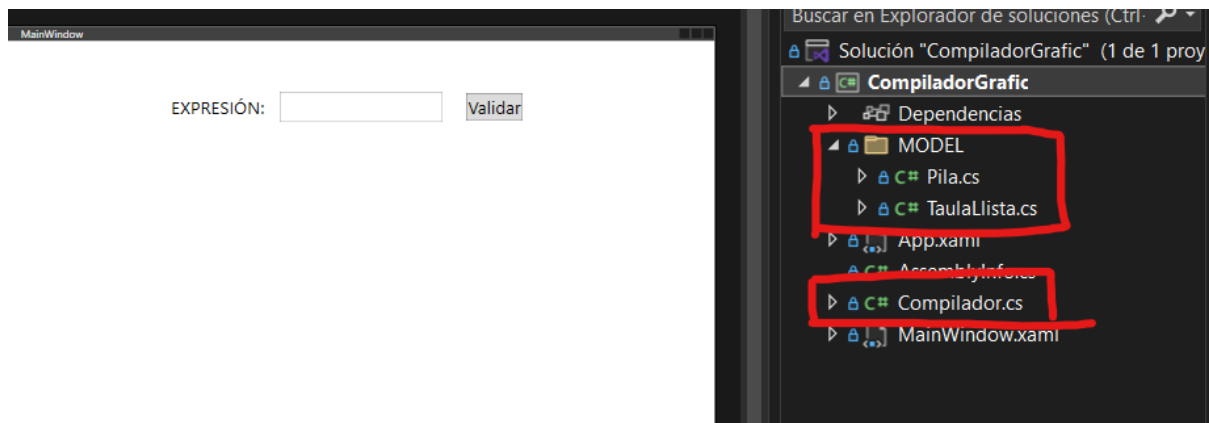
Para asegurarme de que estén todos los cambios actualizados y que no se creen conflictos, hago un git fetch:

```
Usuario@DESKTOP-SGELNL5 MINGW64 ~/source/repos/WS8-BRANQUES-GIT (feature-EntornGrafic)
$ git fetch origin
```

Y por último, el git rebase:

```
Usuario@DESKTOP-SGELNL5 MINGW64 ~/source/repos/WS8-BRANQUES-GIT (feature-EntornGrafic)
$ git rebase origin/develop
Successfully rebased and updated refs/heads/feature-EntornGrafic.
```

Comprobación de la correcta combinación de cambios:



Ahora en la rama de entorno gráfico me sale tanto la carpeta de clases como el compilador. Básicamente lo que hace el rebase es añadir la información combinada sin mostrar los commits de esos cambios. Es como si yo hubiese creado la rama Entorno Gráfico después de haber subido el compilador a develop.

Combinación de Entorno Gráfico a Develop

Ya he acabado todas las partes del proyecto, así que voy a hacer un pull request a develop con los cambios de feature entorno gráfico y luego otro pull request a main con todos los cambios de develop.

The screenshot shows the GitHub 'Comparing changes' interface. At the top, it says 'Comparing changes' and 'Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#) or [learn more about](#)'. Below this, there are two dropdown menus: 'base: develop' and 'compare: feature-EntornGrafic'. A green checkmark and the text 'Able to merge. These branches can be automatically merged.' are displayed. Underneath, there's a section 'Add a title' with the text 'Feature entorn grafic'. Below that is 'Add a description' with a rich text editor showing 'Combinación de cambios del entorno gráfico a develop'. At the bottom, there's a green checkmark icon and the text 'No conflicts with base branch' and 'Merging can be performed automatically.' A green button 'Merge pull request' is visible, along with a link to 'View command line instructions'.

Ya que no hay conflictos, acepto la solicitud y borro la rama.

The screenshot shows a GitHub notification banner. It has a purple icon with a checkmark and the text 'Pull request successfully merged and closed'. Below this, it says 'You're all set — the `feature-EntornGrafic` branch can be safely deleted.' There is a button 'Delete branch' on the right.

The screenshot shows the GitHub 'Branches' page. It has tabs for 'Overview', 'Yours', 'Active', 'Stale', and 'All'. Below the tabs is a search bar with the text 'Search branches...'. Underneath, there's a table with two rows: 'main' and 'develop'. Each row has a copy icon to its right.

Subida final del proyecto

Una vez finalizado el proyecto y con todo subido a develop, toca comprobar el correcto funcionamiento.

MainWindow

EXPRESIÓN:

Validar

Seleccionar Archivo

Expresión	Validación	Resultado
$2*(3+4)-5$	Válido	9
$(8+2)/(5-2*1)$	Válido	3
$3+4-2*(1-7)$	Válido	19
$6/(1+2)*4-3$	Válido	5
$(5+3)*2-8/(4-1)$	Válido	14

Ya que el programa funciona correctamente, ya se puede subir a main con un pull request.

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across](#)

↕

base: main

←

compare: develop

✓

Able to merge.

These branches can be automatically merged.

Add a title

Subida final del programa

No hay errores y eliminamos la rama develop:

✓

No conflicts with base branch

Merging can be performed automatically.

Merge pull request

▼

You can also merge this

Branch

main